Analyzing Pre-Trained Neural Network Behavior with Layer Activation Optimization

Melissa C Phillips
School of Data Science
University of Virginia
Charlottesville, USA
mcp3vj@virginia.edu

Rebecca Stein
School of Data Science
University of Virginia
Charlottesville, USA
rcs8bg@virginia.edu

Taeheon Park
School of Data Science
University of Virginia
Charlottesville, USA
tp8pj@virginia.edu

***Abstract*—Image classification and object recognition with neural networks could have applications in aesthetically-focused branches of the humanities, such as landscape architecture. However, such methods require either the assembly of a massive, domain specific labeled data set or use of network weights initialized on another data set, a technique known as transfer learning. Transfer learning research has established that a pre-trained convolutional neural network (CNN) can achieve high accuracy on new image recognition tasks with relatively few training images. In practice, pre-trained tends to mean pre-trained on ImageNet, the standard dataset for computer vision research. Experiments have shown that the dataset on which a pre-trained model was originally optimized can quantitatively bias it. The goal of this project was to design an experiment to qualitatively analyze how the dataset used to initialize a pre-trained classification system affects its behavior at progressive network layers using feature visualization strategies. We initialized two ResNet-18 CNNs with weights pre-trained on ImageNet and the Places365 dataset, respectively, and fine-tuned them for a new classification task on a landscape image dataset which we collected. Using class activation optimization methods taken from the deep visualization literature, we compared the network filters at several hidden layers and the final output layers. The class activation optimization results show that even at early stages in the networks, their neurons exhibit notably different behavior. Accordingly, we show both that feature visualization techniques can be used to qualitatively study the effect of original training data on transfer learning and, consequently, that the homogeneous use of ImageNet in computer vision experiments may have notable implications for model behavior.**

***Index Terms*—Machine learning, Multi-layer neural network, Image databases**

I. Introduction

When facing an image classification problem with only a small dataset of training images, using a pre-trained convolutional neural network (CNN) to initialize the feature weights before fine-tuning on the data becomes an attractive option. The motivation for the project came from this scenario, where a group working to digitize a seminal work on landscape architecture approached the authors to build a machine learning model to classify landscape images using the approximately 600 images in the book as training data. Working with a highly specialized dataset led to concerns about the easily available pre-trained models (trained on ImageNet) and their efficacy on the dataset. An alternate dataset (Places365) was sought to see how models trained on it compared in their final layers after being fine-tuned on our data. Feature visualization arose as a way to explore how the layers of the CNN were activated as a result of training. Feature visualization is a field concerned with creating semantically intelligible representations of a neural network's learned weights. Past work has applied feature visualization techniques to analyze the transfer learning process. Here we applied the same techniques to understand how the training data itself affects that process. CNNs appear to learn increasingly task-specific representations at each successive layer. Therefore, we hypothesized that the two networks would behave most similarly at the shallowest layers and then diverge relative to depth. In the next section we discuss the research that informs this project.

Many thanks to our sponsor, the Landscape Studies Initiative, Center for Cultural Landscapes, University of Virginia School of Architecture and our advisor, Prof. Raf Alvarado, School of Data Science, UVA

II. Background and Research

A. ImageNet

In 2009, a paper introducing the ImageNet database [1] changed the field of computer vision research. This ambitious project aimed to produce 500 - 1000 high-quality images labeled according to the WordNet database, which groups English words into 'synonym sets' [2]. At its initial release, ImageNet had 3.2 million clean images in its 12 subtrees [1], and today it includes over 14 million images [3]. Each human-labeled image is subject to quality control standards [1], and the scope of the highest level tags is quite broad, ranging from living organisms, such as animals, insects and people, to inanimate objects, such as musical instruments, fabric, and vehicles [3]. Because of the sheer volume of images and their high quality, ImageNet has become popular in computer vision projects, either in its use for pre-training models or for model evaluation. Even a quick search of recent literature produced numerous papers that rely exclusively on the ImageNet database in these ways for projects in fields as diverse as the digital humanities [4], environmental sciences [5], healthcare [6] - [8] and machine learning research [9] - [17].

There has been some recent work examining the heavy reliance on this dataset and potential consequences, such as Tommasi et al.'s "Combining Multiple Cues for Visual

Madlibs Question Answering" where the authors propose using specialized datasets alongside ImageNet to isolate features specific to the domain [18]. In "Citrus Pests and Disease Recognition", Xing et al. raise concerns about overfitting and the waste of computing resources in the widespread use of ImageNet trained CNNs and propose scaled-down architectures for computer vision problems on small datasets [19]. Kornblith et al. consider transfer learning with ImageNet pre-trained networks and find that for more "fine-grained" tasks, the ImageNet pre-trained models do no better than models that were not pre-trained [20]. Additionally, a recent experiment with ImageNet pre-trained CNNs showed that they tend to favor texture over shape when making classification decisions [21]. These concerns resonated with us and helped shape our project.

B. Places

As our initial project dealt with images of landscapes, we looked for a more domain-specific dataset to compare with ImageNet. We found the Places dataset, which was built by labelling images by the place where they occur, rather than the objects within the frame [22]. The dataset building process began by first building an exhaustive list of categories of places, using WordNet to create the Scene UNderstanding (SUN) dataset. From there, these terms were narrowed down to 900 categories, and images were found and labelled by multiple humans to ensure quality control. The Places365-standard dataset is further reduced to 365 categories and each category has between 3,068-5,000 images and the dataset as a whole has over 1.8 million images [22]. Preliminary research on CNNs pre-trained on Places365 show that it can be used for transfer learning in scene classification tasks as well as object detection. Our experiment compares a model pre-trained on Places with one pre-trained on ImageNet.

III. Data Description and Pipeline

The image dataset was compiled by web-scraping certain sites like Flickr, Google Images and the Wikimedia Commons, and downloading from various open collections in ArtStor. The images were manually divided into two categories: Designed Landscapes and Natural Landscapes. Natural Landscapes are those that occur in nature without any human intervention, and Designed Landscapes are those that have been altered by humans, for example planting in rows or other shapes and/or introducing human-built features such as benches or buildings, etc. These images were further divided into Training and Validation sets, with 1936 Designed and 423 Natural Landscape images in the Training set, and 463 Designed and 180 Natural Landscape images in the Validation set.

The images were stored in an allocation on Rivanna, UVA's high-performance computing cluster. Transformations and augmentation of the data occurred as images were loaded into PyTorch's 'dataloader' class. For the Training set, this included resizing each image to 256 pixels, cropping at the center to 224 pixels (the required input to the model), random horizontal flipping, and normalizing each input channel. The images were sent to the model in batch sizes of 4. The only change for the Validation set was that no horizontal flips were made.

Fig. 1. Natural Landscape [30][31]

Fig. 2. Designed Landscape [32][33]

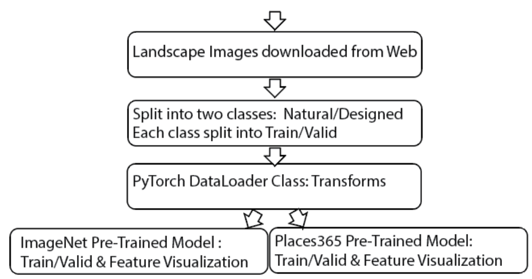

Fig. 3. Data Pipeline

IV. Technical Overview

A. Convolutional Neural Networks and ResNet-18

CNNs have risen in popularity for their use in image recognition tasks after the SuperVision team won the ImageNet Large Scale Visual Recognition Challenge (ILSVRC in 2012 with a deep CNN nicknamed "AlexNet" [23]. One reason they are so effective for the task is because of the use of a filter which processes pixels by including their nearest neighbors, rather than one-by-one. This allows for filters in the earliest layers to pick up simple lines and shapes and for these to be fed into the deeper layers to produce more complex feature maps. Over the years, the ILSVRC has provided the opportunity for teams to try new CNN architectures and see how well they compare against each other. In 2014, the VGG team submitted a CNN with more layers (16-19) and by winning the localization task that year and placing second in the classification task, they showed that deeper networks can improve model outcomes [24].

One drawback to using a network with more layers is that the training process is more challenging, especially as exploding/vanishing gradients become a problem when trying to optimize the network. The creators of the ResNet model noticed this and found that while normalizing the inputs and including normalization layers in the architecture can mitigate some of this, another way to improve accuracy in the very deep networks is by leveraging "residual learning". This means that there are points in the feedforward network where a "shortcut" exists, allowing an identity layer to be passed one or more layers up in the network, ideally making the optimization task for the entire network easier [25]. An ensemble model of ResNet CNNs won the ILSVRC in 2015, and ResNet architecture has become widely used for pre-training networks. The ResNet-18 model is suitable for our problem because it is easily implemented on PyTorch and has already been pre-trained on the ImageNet and Places datasets.

Our pre-trained models achieved similar accuracies on our training set. The details are outlined in Table 1.

TABLE I
ACCURACY OF PRE-TRAINED MODELS ON LANDSCAPE DATASET

Model	**Training**	**Validation**
ImageNet Pre-trained	96%	80%
Places Pre-trained	99%	81%

B. Class Activation Optimization

To obtain a generic visual representation of each neural network's final output layer, we applied class activation optimization [26][27]. During the training of a neural network, layer weights are adjusted to minimize a loss function with respect to a fixed input vector. If we instead take a fixed tensor of weights and define our loss function as the output of a given neuron, we can optimize the input vector to maximally activate some part of the neural network. The resulting vector can be interpreted as a representation of what that part of the network is "looking" for when fed an input. Mathematically, let x be a 3 x 224 x 224 tensor representing a 224 x 224 RGB image, with $x_i \in [0, 1]$. Let y be a 2 x 1 vector where y_1 is the probability that an image is from the "designed landscape" class and y_2 is the probability that an image is from the "natural landscape" class. This can be generalized to any layer Z by letting y be a flattened vector of layer activations that x produces and y_i equal the activation of node i in layer Z. To obtain an optimal image for class or node i, we define a loss function $L(x) = -y_i$ and seek to find:

$$\text{argmin}\, xL(x) \tag{1}$$

We chose to minimize the negative probability rather than maximize the probability because PyTorch has an off-the-shelf method for stochastic gradient descent, but not stochastic gradient ascent. Following results from [26], we initialized x as the mean of all validation set images. This produced a smoother color distribution than random initialization. Based on trial and error, we ultimately chose a .06 initial learning rate with a 25% decay every 100 iterations and ran the optimizer for 500 iterations.

C. Grad-CAM and Grad-CAM++

Grad-CAM stands for Gradient-weighted Class Activation Mapping which is a way to find out which part of an image the CNN classification model sees when it determines the class of an image. It specifically utilizes the gradient information about classes as it goes to the final layer of a CNN to create a rough localization map of the important regions in the image, based on its calculations of the most significant neurons for that class [28]. Applying Grad-CAM to an image shows the heat map of a region of the image that the CNN is using to determine its classification. Although Grad-CAM has provided marked improvements over prior feature visualization strategies in its applicability to any CNN without changes to architecture or re-training, it has some limitations. For example, Grad-CAM's localization performance is reduced when there are multiple occurrences of objects of the same class in an image [29]. Also, the Grad-CAM heat map often fails to fully capture the entire object for single object images. To overcome these limitations, Grad-CAM++ was introduced as a generalized version of Grad-CAM which improves the previously mentioned deficiencies [29]. Since Grad-CAM++ takes an approach where important pixels are weighted and evaluated for how much they factor into the CNN's classification decision, it can produce better performing visualizations with the same amount of computation as Grad-CAM [29]. We use both Grad-CAM and Grad-CAM++ to qualitatively assess the performance of our two models by comparing the results of their application to select images.

V. APPLICATION AND RESULTS

A. Class Activation Optimization

Fig. 4 contains image inputs which maximally activate the Designed Landscape class in the neural network trained on Places365 (left) and ImageNet (right). It is clear that the ImageNet network responds to a wide spectrum of bright colors and defined shapes, while the Places365 dataset responds primarily to muted greens and grays with dispersed patterns.

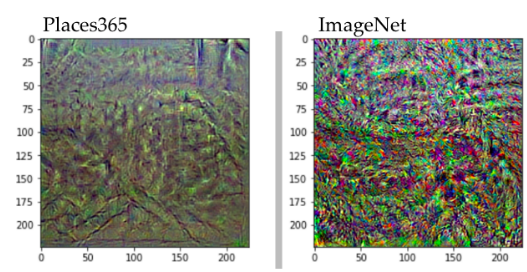

Fig. 4. Designed Landscape Class

The images in Fig. 5 were produced by adding a penalty for the mean value of the maximum pixel value across the three color channels, with the intended effect of suppressing unrealistically saturated pixels. The penalty almost completely desaturated the Places365 image, while some bright colors remain in the ImageNet image, further suggesting that the

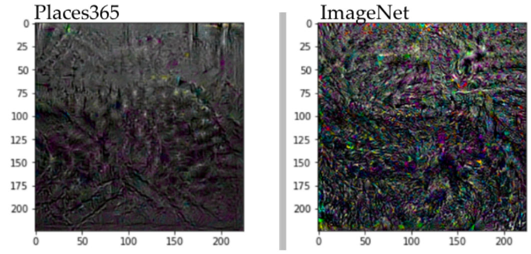

Fig. 5. Designed Landscape Class

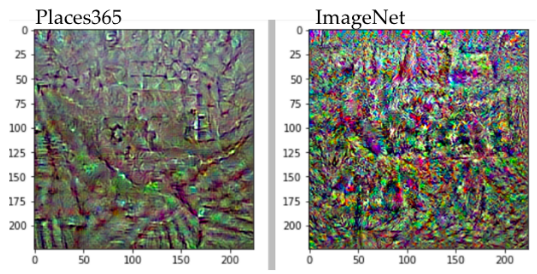

Fig. 6. Natural Landscape Class

neural network trained on ImageNet has a higher sensitivity to color.

Fig. 6 shows images that maximally activated the Natural Landscape node in the softmax layer of the networks trained on Places365 (left) and ImageNet (right). Similar to the results for the Designed Landscape node, we observe that the ImageNet trained network appears more responsive to bright colors and local forms while the Places365-trained network responds to a web-like pattern imposed over the entire image.

Interestingly, when we applied the penalty for mean maximum pixel value to the loss function, we noticed only a very slight change in the images.

We repeated the process using the output from hidden layers 2, 3, and 4 to test whether the network behavior grew more divergent at higher levels. Samples of the images that were optimal with respect to specific nodes are below.

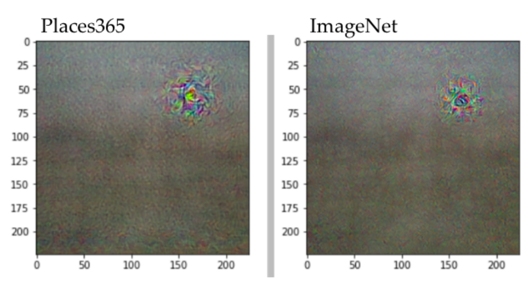

Fig. 7. Layer 2, node 9367

At layers 2 (Fig.7) and 3 (Fig. 8), the ImageNet network appears to focus on tighter regions. In layer 4 (Fig. 9), it is apparent that ImageNet responds to bold colors and more clearly delineated shapes than the Places 365 network.

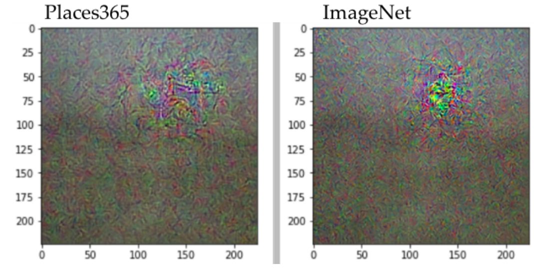

Fig. 8. Layer 3, node 3200

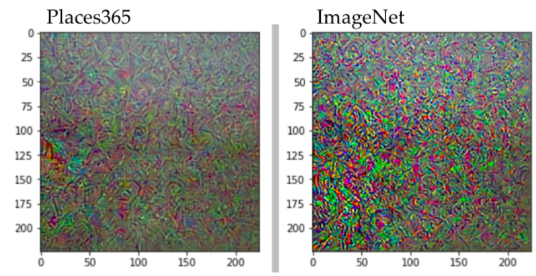

Fig. 9. Layer 4, node 1400

B. Grad-CAM and Grad-CAM++

The Grad-CAM and Grad-CAM++ results were obtained by using the Pytorch module, gradcam. After loading our Resnet-18 models, we applied Grad-CAM functions. We then used the four images in Figs. 8 - 11, two images from each class, to generate Grad-CAM and Grad-CAM++ results.

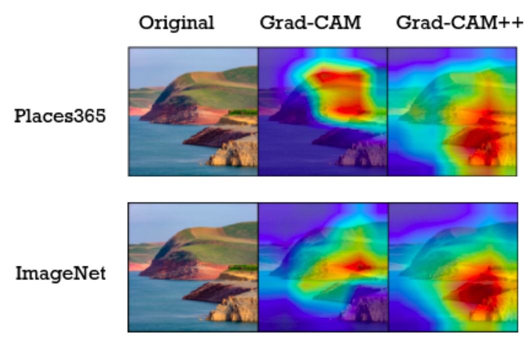

Fig. 10. Natural Landscape Class

Fig. 10 shows the Grad-CAM and Grad-CAM++ results based on the first natural landscape image. For the Grad-CAM, the result from the Places365 model shows the relatively wider heat map on the image, and the location of the heat map is on the mountain ridge. The ImageNet dataset result, however, shows a smaller heat map focused on the bush. The Grad-CAM++ results show that the heat maps are focused on similar locations for both datasets, but the model pre-trained on the Places365 dataset has a a slightly wider view.

Fig. 11 shows the Grad-CAM and Grad-CAM++ results based on the second natural landscape image. The Grad-CAM result from the Places365 model shows a wide heat map on the

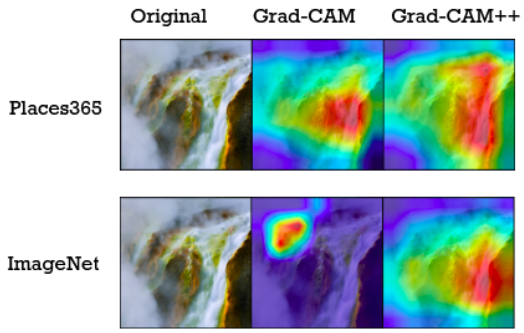

Fig. 11. Natural Landscape Class

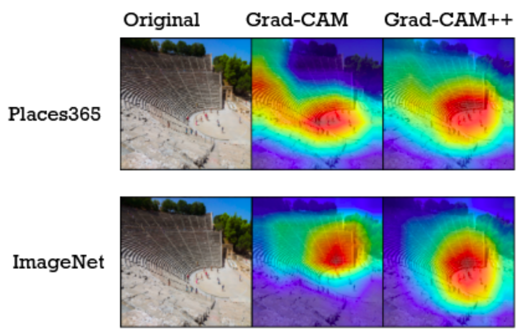

Fig. 12. Designed Landscape Class

image, and the focus of the heat map is on the entire waterfall. The ImageNet model result shows a much smaller heat map which is focused on a small, angular portion of the landscape. The Grad-CAM++ results show heat maps that are focused on similar locations for both datasets, but the model pre-trained on the Places365 dataset focuses on the entire stream of the waterfall, while the ImageNet model only focuses on a portion of the waterfall.

Fig. 12 shows the Grad-CAM and Grad-CAM++ results based on the first designed landscape image. These results show that the Places365 model is focused on broader subsets of the images than the ImageNet model is. The ImageNet model is "looking at" one specific portion of the amphitheater image, but the Places365 model includes the base of the amphitheatre as well as some of the stairs in its scope.

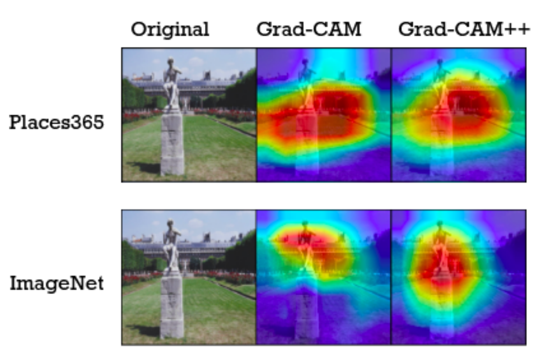

Fig. 13. Designed Landscape Class

Fig. 13 shows the Grad-CAM and Grad-CAM++ results based on the second designed landscape image. Again in these results, we see that the Places365 model has more of the image in view when making classification decisions than the ImageNet model. Both models seem to heavily use the statue to make the determination, but the Places365 model looks lower and takes more of the background into account while the ImageNet model seems to focus almost exclusively on the statue.

For all four images, the Places365 pre-trained model produces broader heat maps thus taking into account more features in the image, while the ImageNet pre-trained model shows smaller heat maps and seems to focus more narrowly on certain image details.

VI. Conclusion and Future Work

Overall, our experiment reveals that feature visualization strategies can be effective ways to learn how the pre-training of a model affects its behavior. The class activation optimization experiments and Gradient-weighted Class Activation Mapping results revealed qualitatively significant differences between the features each original training set produced. These are summarized in Table II.

TABLE II
Summary of Results

Visualization	ImageNet Model	Places365 Model
Class Activation Optimization	-Responded to bolder colors -Responded to defined shapes	-Responded to green hues -Responded to repeating block or fractal patterns
GradCAM	-Produced smaller heat maps	-Produced wider heat maps

The ImageNet trained neural network responded to bolder colors and more defined shapes than the Places 365 network, which appeared to respond to green hues and block or fractal patterns which repeated across the entire image. As a result of the mean-value penalty we applied to the class activation optimization experiment, we made the unexpected observation that color appears to influence the activation of the natural landscape softmax node more than the designed landscape softmax node.

Based on the results of Gradient-weighted Class Activation Mapping, we can see that the Imagenet pre-trained model is producing smaller heatmaps, which we hypothesize could be due to the ImageNet database's design as providing labels for objects in images. Conversely, the Places365 pre-trained model produces wider heat maps, which again seems to follow from the dataset's specialization in providing labels for background images.

Knowing that the CNNs behave differently based on the weights obtained from pre-training raises a concern about the widespread use of ImageNet for pre-training in computer vision research. Although the models produced often achieve high accuracy on their given task, it could be that another pre-training dataset could produce changes to the network that might be advantageous if only it had been explored. The Places365 dataset offers another option for pre-training and perhaps new image datasets will continue to be developed to create more options in the field of computer vision.

REFERENCES

[1] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. "ImageNet: A Large-Scale Hierarchical Image Database." In 2009 IEEE Conference on Computer Vision and Pattern Recognition, 248–55, 2009. https://doi.org/10.1109/CVPR.2009.5206848.

[2] "WordNet — A Lexical Database for English." Accessed March 28, 2020. https://wordnet.princeton.edu/.

[3] "ImageNet." Accessed March 28, 2020. http://www.image-net.org/.

[4] Linda Studer, Michele Alberti, Vinaychandran Pondenkandath, Pinar Goktepe, Thomas Kolonko, Andreas Fischer, Marcus Liwicki, and Rolf Ingold. "A Comprehensive Study of ImageNet Pre-Training for Historical Document Image Analysis." In 2019 International Conference on Document Analysis and Recognition (ICDAR), 720–25, 2019. https://doi.org/10.1109/ICDAR.2019.00120.

[5] J. Christopher D. Terry, Helen E. Roy, and Tom A. August. "Thinking like a Naturalist: Enhancing Computer Vision of Citizen Science Images by Harnessing Contextual Data." Methods in Ecology and Evolution 11, no. 2 (2020): 303–15. https://doi.org/10.1111/2041-210X.13335.

[6] Andres Diaz-Pinto, Sandra Morales, Valery Naranjo, Thomas Köhler, Jose M. Mossi, and Amparo Navea. "CNNs for Automatic Glaucoma Assessment Using Fundus Images: An Extensive Validation." BioMedical Engineering OnLine 18, no. 1 (March 20, 2019): N.PAG-N.PAG.

[7] K. Shaga Devan, P. Walther, J. von Einem, T. Ropinski, H. A. Kestler, and C. Read. "Detection of Herpesvirus Capsids in Transmission Electron Microscopy Images Using Transfer Learning." Histochemistry and Cell Biology 151, no. 2 (February 2019): 101–14. https://doi.org/10.1007/s00418-018-1759-5.

[8] Marit Lucas, Esmee I.M.L. Liem, C. Dilara Savci-Heijink, Jan Erik Freund, Henk A. Marquering, Ton G. van Leeuwen, and Daniel M. de Bruin. "Toward Automated In Vivo Bladder Tumor Stratification Using Confocal Laser Endomicroscopy." Journal of Endourology 33, no. 11 (November 2019): 930–37.

[9] Deepak Mittal, Shweta Bhardwaj, Mitesh M. Khapra, and Balaraman Ravindran. "Studying the Plasticity in Deep Convolutional Neural Networks Using Random Pruning." Machine Vision and Applications 30, no. 2 (March 2019): 203–16. https://doi.org/10.1007/s00138-018-01001-9.

[10] Er-Yang Huan, and Gui-Hua Wen. "Multilevel and Multiscale Feature Aggregation in Deep Networks for Facial Constitution Classification." Computational and Mathematical Methods in Medicine 2019 (December 20, 2019): 1–11. https://doi.org/10.1155/2019/1258782.

[11] Lin Guo, and Shiyin Qin. "High Precision Detection of Salient Objects Based on Deep Convolutional Networks with Proper Combinations of Shallow and Deep Connections." Symmetry 11, no. 1 (December 21, 2018): 5. https://doi.org/10.3390/sym11010005.

[12] Connor Amorin, Laura M. Kegelmeyer, and W. Philip Kegelmeyer. "A Hybrid Deep Learning Architecture for Classification of Microscopic Damage on National Ignition Facility Laser Optics." Statistical Analysis Data Mining 12, no. 6 (December 2019): 505–13.

[13] Zhiyong Wei, and Fengling Wang. "Adaptive Cascade Single-Shot Detector on Wireless Sensor Networks." EURASIP Journal on Wireless Communications and Networking 2019, no. 1 (December 2019): 150. https://doi.org/10.1186/s13638-019-1440-2.

[14] Yang You, Zhao Zhang, Cho-Jui Hsieh, James Demmel, and Kurt Keutzer. "Fast Deep Neural Network Training on Distributed Systems and Cloud TPUs." IEEE Transactions on Parallel Distributed Systems 30, no. 11 (November 2019): 2449–62.

[15] Wei Zhang, Ping Tang, and Lijun Zhao. "Remote Sensing Image Scene Classification Using CNN-CapsNet." Remote Sensing 11, no. 5 (February 28, 2019): 494. https://doi.org/10.3390/rs11050494.

[16] Rizanne Elbakly, and Moustafa Youssef. "The StoryTeller: Scalable Building- and AP-Independent Deep Learning-Based Floor Prediction." Association for Computing Machinery, March 18, 2020. http://doi.org/10.1145/3380979.

[17] Victor-Alexandru Darvariu, Laura Convertino, Abhinav Mehrotra, and Mirco Musolesi. "Quantifying the Relationships between Everyday Objects and Emotional States through Deep Learning Based Image Analysis Using Smartphones." Association for Computing Machinery, March 18, 2020. https://doi.org/10.1145/3380997.

[18] Tatiana Tommasi, Arun Mallya, Bryan Plummer, Svetlana Lazebnik, Alexander C. Berg, and Tamara L. Berg. "Combining Multiple Cues for Visual Madlibs Question Answering." International Journal of Computer Vision 127, no. 1 (January 2019): 38–60. https://doi.org/10.1007/s11263-018-1096-0.

[19] Shuli Xing, Marely Lee, and Keun-kwang Lee. "Citrus Pests and Diseases Recognition Model Using Weakly Dense Connected Convolution Network." Sensors 19, no. 14 (July 19, 2019): 3195. https://doi.org/10.3390/s19143195.

[20] Simon Kornblith, Jonathon Shlens, and Quoc V. Le. "Do Better ImageNet Models Transfer Better?" In 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2656–66, 2019. https://doi.org/10.1109/CVPR.2019.00277.

[21] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A. Wichmann, and Wieland Brendel. "ImageNet-Trained CNNs Are Biased towards Texture; Increasing Shape Bias Improves Accuracy and Robustness." ArXiv:1811.12231 [Cs, q-Bio, Stat], January 14, 2019. http://arxiv.org/abs/1811.12231.

[22] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. "Places: A 10 Million Image Database for Scene Recognition." IEEE Transactions on Pattern Analysis and Machine Intelligence 40, no. 6 (June 2018): 1452–64. https://doi.org/10.1109/TPAMI.2017.2723009.

[23] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, et al. "ImageNet Large Scale Visual Recognition Challenge." International Journal of Computer Vision 115, no. 3 (December 1, 2015): 211–52. https://doi.org/10.1007/s11263-015-0816-y.

[24] Karen Simonyan, and Andrew Zisserman. "Very Deep Convolutional Networks for Large-Scale Image Recognition." ArXiv:1409.1556 [Cs], April 10, 2015. http://arxiv.org/abs/1409.1556.

[25] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep Residual Learning for Image Recognition." ArXiv:1512.03385 [Cs], December 10, 2015. http://arxiv.org/abs/1512.03385.

[26] Anh Nguyen, Jason Yosinski, and Jeff Clune. "Multifaceted Feature Visualization: Uncovering the Different Types of Features Learned By Each Neuron in Deep Neural Networks," n.d., 22.

[27] Jason Yosinski, Jeff Clune, Anh Nguyen, Thomas Fuchs, and Hod Lipson. "Understanding Neural Networks Through Deep Visualization," n.d., 12.

[28] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. "Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization." International Journal of Computer Vision 128, no. 2 (February 2020): 336–59. https://doi.org/10.1007/s11263-019-01228-7.

[29] Aditya Chattopadhyay, Anirban Sarkar, Prantik Howlader, and Vineeth N. Balasubramanian. "Grad-CAM++: Improved Visual Explanations for Deep Convolutional Networks." 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), March 2018, 839–47. https://doi.org/10.1109/WACV.2018.00097.

[30] Myrabella. "La Plage de Bussaglia (Serriera, Corse-Du-Sud, France)". August 20, 2013. Own work. https://commons.wikimedia.org/wiki/File:Bussaglia_plage_Serriera.jpg. Licensed under CC BY-SA 3.0.

[31] Diliff. "A Panoramic View from Glacier Point in Yosemite National Park in California, United States". May 16, 2013. Own work. https://commons.wikimedia.org/wiki/File:Glacier_Point_at_Sunset,_Yosemite_NP,_CA,_US_-_Diliff.jpg. Licensed under CC BY-SA 3.0.

[32] Dietmar Rabich. "Autumn in the Börnste Hamlet, Kirchspiel, Dülmen, North Rhine-Westphalia, Germany". November 1, 2015. Own work. https://commons.wikimedia.org/wiki/File:D\%C3\%BClmen,_B\%C3\%B6rnste,_Waldweg_--_2015_--_4649.jpg. Licensed under CC BY 4.0.

[33] Renardo la Vulpo. "Mündung Des Fuchsluger Bachs in Die Prien in Aschau Im Chiemgau". April 5, 2016. Self-photographed. https://commons.wikimedia.org/wiki/File:Aschau_im_Chiemgau,_Fuchsluger_Bach_und_Prien,_1.jpeg. In the Public Domain, CC0.